**数据结构**

|  |  |
| --- | --- |
| 创建元组 | tup1 = 4,5,6 或 tup2 = (6,7,8)  注意：  ·切片时包含start位置元素，但不包含stop位置元素  ·若省略start stop表示序列的首尾位置 |
| 创建嵌套元组 | tup1= (4,5,6),(7,8) |
| 将序列或迭代器转换为元组 | tuple([1,0,2])  Step的应用：   |  |  | | --- | --- | | 每隔一个元素选取 | List1[::2] | | 字符串倒序 | Str1[::-1] | |
| 连接元组 | tup1 + tup2 |
| 元组解包 | a,b,c = tup1 |

**元组应用 字典（HASH MAP）**

|  |  |
| --- | --- |
| 创建字典 | Dict1 = {‘key1’:’value1’,2:[3,2]} |
| 从序列中创造字典 | Dict(zip(keyList,valueList)) |
| 获取元素  插入和修改元素 | Dict1[‘key1’]\*  Dict1[‘key1’]=’newValue’ |
| 获取元素（带默认值） | Dict1.get[‘key1’,defaultValue]\*\* |
| 检查键是否存在 | ‘key’ in dict1 |
| 删除元素 | del dict1[‘key1’] |
| 获取键列表 | Dict1.keys()\*\*\* |
| 获取值列表 | Dict1.values()\*\*\* |
| 更新值 | Dict1.update(dict2) |

|  |  |
| --- | --- |
| 交换元素 | b,a = a,b |

\* 如果key不存在，会出现‘KeyError’异常

\*\* 用‘get()’函数获取元素，若元素不存在返回‘None’

\*\*\* 返回键和值的列表是相同的顺序。然而，这个顺序没有特殊意义，可能没有排序

**列表**

列表是Python中一维的，长度可变的序列对象。

列表是可变的（例如内容可能会被改变），

其元素可以是任意类型。

|  |  |
| --- | --- |
| 创建列表 | list1 = [1,’a’,3] 或  list1 = list(tup1)  字典键的合法形式  ·键必须是一种不可变类型（int，float，string）或元组（元组中的所有元素形式也必须相同）  ·这里的术语是‘hashability’，检查对象是否可哈希，例如hash(‘this is string’)，而hash([1,2])会 出错 |
| 连接列表 | list1 + list2 或  list1.extend(list2) |
| 列表末尾添加元素 | list1.append(‘b’) |
| 在指定位置插入元素 | list1.insert  (posIdx,’b’)  **集合**  ·一个集合是一组无序的不重复的元素  ·你可以把他们当作只有键的字典   |  |  | | --- | --- | | 创建集合 | Set([3,6,3])或{3,6,3} | | 检测子集 | Set1.issubset(set2) | | 检测父集 | Set2.issusuperset(set2) | | 检测集合包含相同内容 | Set1==set2 | |
| 在指定位置删除元素 | valueAtIdx = list1.pop(popIdx) |
| 移除列表第一个元素 | list1.remove(‘a’) |
| 检查是否属于列表 | 3 in list1 => True |
| 排序列表 | list1.sort()  **集合操作：**   |  |  | | --- | --- | | 并集（或） | Set1|set2 | | 交集（和） | Set1&set2 | | 差集 | Set1- set2 | | 对称差(异或) | Set1^set2 | |
| 按用户要求排序列表 | list1.sort(key = len) |

1. 使用“+”进行列表连接代价很高，因

为必须创建新列表并在其上复制对象。因

此，推荐使用extend()。

1. Insert函数比append函数的计算的代价更高。
2. 从列表里查找一个元素比在字典（dicts）和集合（sets）中

慢，因为在列表中查找，Python进行线性扫描，而在字典和集合

中（基于哈希表）查找可以在常数时间内完成。

**内置二分模块（bisect）**

·在有序的列表中实现二分查找和插入

·bisect.bisect方法用来查找位置，bisect.insort方法用来在此位置上插入元素

警告：bisect模块函数不检查列表是否排序，

这样做在计算上会很昂贵。因此，用于没有

排序的列表不会发生错误，但是可能会

得到不正确的答案。

**序列类型的切片**

序列类型包括’str’,’array’,’tuple’,’list’等等

|  |  |
| --- | --- |
| 语法 | List1[start:stop] |
|  | List1[start:stop:step](step表示步长) |